

# Proyecto: e-Commerce con React

Construye un e-commerce (sitio de comercio electrónico) con React a partir de consumir una API de un backend genérico proporcionado.

**El diseño es libre y esta vez SÍ se tomará en cuenta.**

## API

Se utilizará una API local basada en JSON (mock) para el desarrollo de la tienda. Por lo que deberás tenerla descargada en tu equipo y corriendo a modo local para poder realizar el proyecto.

La URL con el backend a usar y la documentación es:

<https://github.com/warderer/json-server-jwt>

## REQUERIMIENTOS FUNCIONALES

Se recomienda realizar el proyecto con una aproximación por fases, añadiendo nuevas características en cada iteración.

Última actualización:

29-01-2023

Por César Guerra (cesar@cesarguerra.mx)

## MVP FASE 1: Ver y Consultar Productos

En esta etapa nos centraremos en arrancar el proyecto y poder consumir la API, con el objetivo de mostrar la información más relevante de una tienda: Los productos y detalles de los mismos.

### Características Generales:

- Mostrar productos
- Mostrar detalles de producto
- Buscar productos

### Funcionalidades:

- Cuando entro al home puedo ver todos los productos.
- El componente NavBar debe estar a la vista en toda la app.
- Siempre hay una barra de búsqueda en la NavBar para hacer búsquedas por productos.
- Si doy click a un producto, puedo ver su detalle completo en una URL / ruta única para este producto.
- Dentro del detalle individual de un producto, se muestra el botón "comprar". En esta primera etapa no tendrá lógica de programación dicho botón, pero debe ser considerado para fines de diseño y más adelante hacerlo funcionar.

### Páginas/Vistas que recomendamos crear:

- Productos (Puede ser la Home)
- Producto Individual

**Skills:** APIs, Axios, useState, useEffect, react-router, eventos, useContext (navBar).

## MVP FASE 2: Inicio de Sesión y Registro

En esta etapa nos centraremos en poder consumir la API de inicio de sesión de usuario y realizar un formulario de registro. Así como realizar renderizado condicional en base a si estamos o no autenticados (es decir, la interfaz puede presentar ligeros cambios dependiendo el caso).

### Características Generales:

- Debe tener signup (registro)
- Debe tener login (inicio de sesión)
- Al iniciar sesión, debe distinguir entre usuario de tipo ADMIN y usuario de tipo CUSTOMER

### Funcionalidades CUSTOMER:

#### **Sin Usuario / Con Usuario** (*sin iniciar sesión / con sesión iniciada*)

- Cuando entro al home puedo ver todos los productos
- Siempre hay una barra de búsqueda en la navbar para hacer búsquedas por productos.
- Si doy click a un producto, puedo ver su detalle completo en una URL / ruta única para este producto.

#### **Sin Usuario (sesión no iniciada):**

- En la navbar, del lado derecho, me sugiere hacer signup o login.
- La vista de signup, y la vista de login, son urls diferentes ( /login /signup )

- Hay un botón comprar en la vista individual del detalle de cada producto, pero este está desactivado y debes mostrar (como desees) una invitación a registrarte o iniciar sesión.

## **Con Usuario (sesión iniciada):**

- En la navbar se muestra el nombre de usuario
- Dentro del detalle individual de un producto, se muestra el botón "Comprar" habilitado.

## **Páginas/Vistas que recomendamos crear:**

- Login (Iniciar sesión)
- Signup (Registro)

**Skills:** useContext, Autenticación con JWT, localStorage, Manejo de formularios y eventos

## MVP FASE 3: Crear productos como Administrador

Dotaremos a la aplicación de la capacidad de distinguir a un usuario normal de un usuario administrador. El usuario administrador podrá crear nuevos productos.

### Funcionalidades ADMIN:

- Debe tener todas las funcionalidades del CUSTOMER.
- Puede dar de alta nuevos productos.
- Colocar un enlace en la Navbar que le permita cargar productos.

### Páginas/Vistas que recomendamos crear:

- Alta de producto (formulario)

**Skills:** useContext, Autenticación con JWT, localStorage, manejo de formularios y eventos.

**Importante:** La API maneja 2 roles de usuario "CUSTOMER" y "ADMIN". Necesitas logearte con una cuenta de ADMIN para qué puedas cargar el producto.

Como tip, puedes crear con Postman/Insomnia un usuario y añadir la propiedad role al body con el contenido:

```
"role": "ADMIN",
```

Cuando se crea un producto, su **category** deberá ser alguna de las siguientes: "Books", "Movies", "Music", "Games", "Electronics", "Computers", "Home", "Garden", "Tools", "Grocery", "Health", "Beauty", "Toys", "Kids", "Baby", "Clothing", "Shoes", "Jewelery", "Sports", "Outdoors", "Automotive", "Industrial"

## MVP FASE 4: Carrito de Compra

Con ayuda de Context API (useContext) creamos un carrito de compras en nuestra Navbar qué muestre los productos qué agreguemos en la tienda, actualizar sus cantidades/eliminarlos y calcule el total.

### Funcionalidades:

- Agregar la funcionalidad del carrito de compras (agregar al carrito).
- Los productos pueden agregarse al carrito desde la vista individual (detalle) del producto.
- El carrito puede mostrarse mediante un drop-down o una página completa donde muestres todos los productos que se agregaron al carrito.
- En el carrito puedes eliminar los productos.
- En el carrito puedes cambiar la cantidad elegida del producto
- Debes mostrar el precio total (la suma de los precios de todos los productos, según cantidades elegidas).

### Páginas/Vistas que recomendamos crear:

- Ninguna, crear un componente de carrito de compra qué se renderice en el Navbar o una alternativa similar.

**Skills:** useContext, Autenticación con JWT, localStorage, Eventos.